ACEEE Int. J. on Recent Trends in Engineering and Technology, Vol. 10, No. 2, Jan 2014

Breaching of Ring Oscillator Based Trojan Detection and Prevention in Physical Layer

Ameya Nayak, Kang Yen, and Jeffrey Fan
Florida International University
Email: {anaya005, yenk, fanj}@fiu.edu

***Abstract*— Trojan insertion has been made difficult in modern communications in the recent years, due to extensive research work in the direction to protect Integrated Circuits (ICs). Several Trojan detection techniques have been developed to prevent the destruction caused by malicious insertion of Trojan in physical layer, making the process of Trojan insertion much more difficult. In this paper, we highlight our major findings in terms of innovative Trojan design that can easily evade existing Trojan detection approaches based on side-channel analysis. We propose a design that makes Trojan undetectable for known defense benchmarks and during functional testing. We demonstrate our Trojan model and validate the results on a known defense mechanism. We also present a novel detection mechanism for the same proposed Trojan model. The experimental results are presented using Xilinx Place and Route characteristics, in particular, the Floorplanner tool to identify the Trojan and address such problems.**

***Index Terms*— Hardware Trojan, Ring Oscillator, Trojan detection**

I. Introduction

The digital world has advanced to a great extent. Most of the in-house work is being outsourced for several reasons considered valid by the industry. One of the main reasons for this is maximizing profits and achieving economy whilst not compromising with the quality. However, outsourcing the manufacture of ICs involves the risk of a breakdown in security at any stage. Malicious inclusion of a Trojan can lead to damages in the functioning of the chip or leakage of confidential information to an adversary among other possibilities, threatening the circuits bound for sensitive applications. Various detection techniques have hence been developed to prevent damages caused by Trojans [1]. These Trojan detection mechanisms are generally classified as: side-channel signal analysis [1] [2] [6] [7] [8]. Trojans can be detected based on side-channel analysis by measurement of various circuit parameters, including power based analysis [2], current analysis [6], and delay based analysis [7] [8]. There are many parameters affecting these side-channel parameters such as measurement noise, process variations and environmental changes. In [3], the Hardware Trojan Horse (HTH) mechanism is proposed based on at-speed delay measurement technique which measures delay using long, unique signatures for authentication purposes. Another mechanism in [4] presents a Design-for-Trust (DFTr) technique that can successfully detect a Trojanized section of any circuit by adding a small amount of logic that reconfigures functional paths into Trojan detecting Ring Oscillator (RO) paths. However, hardware Trojans that can be masked by these parameters causing any circuit to malfunction could be potentially designed.

Trojans can be designed according to the following rules: (a) Trojans have an insignificant effect on parameters of the IC such as transient current, delay or power, (b) Trojans are activated under very specific

DOI: 01.IJRTET.10.2.28

conditions (e.g. connection to low switching probability nets) are unlikely to be activated by random or functional stimuli, and (c) Trojans do not cause change to the original circuit's functionality, but adding extra gates to the circuit will cause leakage of the confidential information to an adversary without being detected by the aforementioned approaches. During hardware trust and security assessment, the information on attack models, the attacker intentions and capabilities are unknown a priori. Moreover, the attacker can adapt to the above mentioned defense mechanisms. Consequently, the defenses cannot assume an attack model and hence the traditional Very Large Scale Integration (VLSI) testing based approaches do not provide adequate assurances.

In this paper, we exploit the weakness of the design characteristics posed by such design mechanism. We present a Trojan design that mal-functions the circuit and will still not be detected by the RO technique. In addition, we provide a novel solution to remedy this problem. The organization of paper is as follows: Section II presents our proposed Trojan model which can be inserted in RTL design phase; Section III describes RO Trojan detection technique.

II. Related Work

Malicious changes can be made at any phase of an IC design such as specification, design, fabrication, testing, and packaging. These changes can be as small as adding a trigger and payload in the critical part of the circuit. Jin and Makris [7] proposed a behaviour-oriented categorization of Trojans which helps not only in constructing Trojan models but also lowers the cost of testing. In [4] circuit paths are reconfigured into functional RO paths by adding a small logic that detects inserted Trojans by measuring changes in RO frequencies. A single RO can be designed that simultaneously gives delay information from multiple gates of a design. However, according to the author, this proposed DFTr technique with 100% gate coverage does not guarantee that all attacks are detected. Another implementation of RO based approach has been conducted in which a ring oscillator network (RON) is used to detect fluctuations in characteristic frequencies caused by malicious modifications in the circuit under authentication. Suh and Devdas [5] have presented low cost authentication of various PUF designs that exploit delay characteristics of wires and transistors. Karri et.al [9] proposed taxonomy that can be used to build hardware security modules to protect and prevent future attacks. Zhang et.al [12] presented novel on-chip RON distributed architecture across the entire chip to verify if the chip is Trojan-free. This structure eliminates the issues of measurement noise and compensates the impact of process variations. The same authors implemented the RON structure in an application specific Integrated Circuit (ASIC) and found that the architecture successfully detects the Trojans placed within the circuit. However this method is unsuccessful in detecting a Trojan. In other words, if the characteristic frequency measurement does not show any deviation from the standard frequencies of a Trojan-free IC, then the Trojan remains undetected. This drawback in the design was exploited to breach the RON based Trojan detection system.

III. Proposed Trojan Model

Hardware Trojans can be designed to cause a secret key to be leaked or simply destroy the function of the chip by causing malfunctioning of a specific unit by activating a counter for a certain period of time. Any Trojan can be defined by two primary elements: (i) Trigger and (ii) Payload. A Trojan *trigger* is the activation mechanism that causes a malfunction of the original circuitry. *Payload* defines the event that occurs once a trigger has been activated. Our Trojan design will cause the circuit to malfunction and give incorrect outputs when it is triggered.

Fig. 1 shows a simple block design of our Trojan model. The figure is a simple illustration of our idea that by applying a trigger to any Circuit under Test, a specific event can be triggered at a specified time and its payload can change the desired output at that instance. The circuit shows a 4-bit adder consisting of a RO internally (not shown). The hardware Trojan taxonomy proposed in [1] has classified several Trojans based on five attributes- Insertion phase, abstraction level, activation mechanism, effects, and location. Our Trojan is inserted in the design phase at the Register Transfer Level (RTL) and will be triggered externally with a user input. The model is placed in I/O and intended to change the functionality of the circuit.

The payload can be connected to the original circuit (explicit Trojan), keeping the Trojan size relatively small and placing the Trojan outside the RO and right before the output nets. The output from the counter (which is set to 1) is sent to the OR gate and the second input to it comes from the Adder. Hence, the output of the

corresponding ‘benchmark circuit’ would be ‘1111’ at a particular or multiple input combinations as defined by the counter circuit in the design.

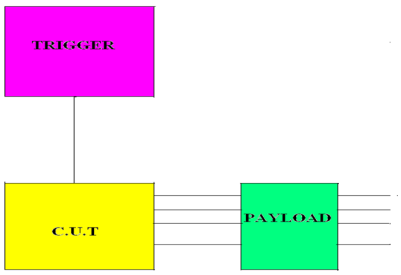

Figure 1. General Block Representation of Trigger and Payload Mechanism for Trojan Insertion

IV. RO TROJAN DETECTION TECHNIQUE

An accepted Trojan Detection method which is a Physically Unclonable Function (PUF) based method was proposed by the authors in [4]. An RO Network is embedded into a circuit and acts as a ‘monitoring architecture’ type Trojan detector. A typical RO consists of multiplexers and several original gates. This RO executes itself in two modes namely, ‘test’ mode and ‘functional’ mode. The RO executes its task of detecting any malicious Trojan when the test mode is activated; while in the functional mode, the RO is disabled and the circuit will function as expected.

A certain set of inputs, which activates the RO lies with the chip owner. When one of the relevant test vectors is given, the multiplexers, inverters and original gates will form a loop, resulting in the activation of the RO and causing it to oscillate. Its frequency will then be calculated using a counter and this particular feature serves as a mark to differentiate between Trojan-free and Trojan-inserted ICs. Since addition of any additional gates that perform malicious functions may cause a change in frequency, the frequencies of the Trojan-free & Trojan-inserted ICs is bound to be different.

Based on the RO’s sensitivity to different variations (measurement noise, process, environmental variations, etc), if the difference is beyond the permitted tolerance value, then the presence of a Trojan in the IC is confirmed and if the difference is within the permitted tolerance, then the IC can be considered Trojan-Free [4]. Fig. 2 shows the different gates within the circuit. It consists of four XOR gates in all and two AND gates along with one inverter. At a certain test vector, when the test mode is enabled, the path that is highlighted (dark dotted) gets activated.

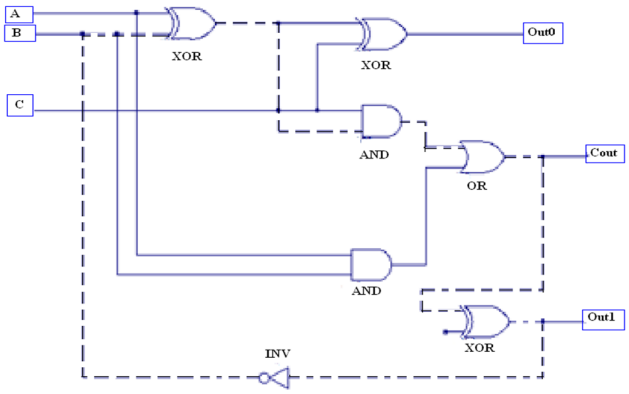

Figure 2. Ring Oscillator Embedded in a Circuit

The loop as indicated by dotted line in Fig. 2 is termed as Ring Oscillator. When one of the inputs to an XOR gate is 1, XOR gate will act like an inverter, which activates the RO. The output from the XOR gate is passed to the inverter and is given as an input to B (where the initial input as 1). While the input combinations could be many, the frequencies generated by RO loop are going to turn out to be the same. The idea is to insert a Trojan in the area that is not covered by the RO. Thus, a Trojan could be inserted into any portion that is not within the dotted portion of Fig. 2. This can cause a change in the output namely, Out0, Cout and Out1.

The RO based design for trust technique detects any malicious intrusions by change in frequency of the RO [4]. Therefore, using this technique, it has been shown that the frequency between a Trojan-free and Trojan-inserted circuit is different. Therefore, the RO design identifies Trojans that may be embedded within the circuit. However, Trojans present outside RO based design loop will remain undetected. Our proposed model as described earlier has been successful in breaching this RO design mechanism.

V. Experimental Results And Analysis

We implemented our Trojan model into a well-established Ring Oscillator Network (RON) architecture [12]. RON based design hardening approaches [12, 13] were proposed for hardware Trojan detection through RO frequency change. The architectural design is shown in Fig. 3. In this architecture, several RO's are embedded into the design of a circuit. Select bits will enable one or multiple of such RO's whose output is provided to a counter which basically counts the frequency generated by each of these RO's. This output can be further analyzed using Data Analysis. The RO based design detection approaches mainly fall into two categories: (1) to secure the design by dynamically configuring circuit paths into the design to monitor any unwanted changes, (2) to insert an addition RON to detect voltage drops due to extra Trojan circuitry. There are certain parameters such as measurement noise, process variations, and environmental variations that have an impact on such frequencies generated by RO. Hence, a tolerance value must be specified when measuring the frequency of each RO loop. There have been several studies considering the effects of these process variations, specifically inter-die and intra-die variations [14]. Several ring oscillators were implemented on multiple FPGAs

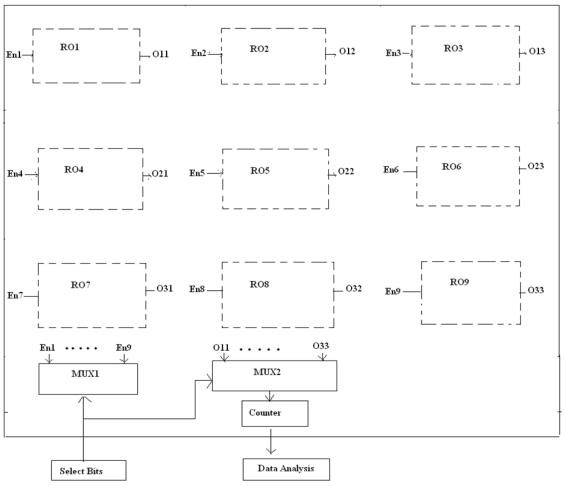

Figure 3. Design architecture of RON embedded circuit

(Xilinx Spartan3E S500) and impact of both inter-die and intra-die variations were studied. The maximum value of both inter-die and intra –die variation was found to be 6.6%. According to the ring oscillator frequency monitoring detection technique, if this value exceeds 6.6%, a Trojan is detected successfully.
Our Trojan model when placed, lie within this range (as shown in Table 1), thus making it difficult to detect by ring oscillator frequency monitoring technique.
When the RO frequencies were tested in the FPGA based framework, the measured RO frequency changes for different inserted Trojans were found to be within the tolerance of process induced parameter variations. The trigger condition of the Trojan only adds load capacitance to some internal nodes of the circuit. Even if all paths and gates are covered by some ROs, the Trojans can still be inserted without much affecting the delay, thereby evading the ASIC framework. The inserted Trojan was not detected by the frequency based RON technique due to the minimal impact on path delay. As the trigger condition is rare, the Trojan will not be detected in functional testing as well.
We use different benchmark circuits that are embedded with ring oscillators throughout their design. The details of these benchmarks circuits can be found in [10]. Table 1 contains the data about the different benchmark circuits with the corresponding number of inputs. We embedded each of these benchmark circuits

with Ring Oscillators and measured their corresponding frequencies which can be represented by counter values. We, then inserted our proposed Trojan model consisting of a counter and OR gates into each of these benchmark circuits are recorded the counter values. The counter values in the last column represent the frequencies for each of the RO benchmark circuits after Trojan insertion. The counter values measures the cumulative frequency for each of the RO loops. We observed that the counter value remain almost unchanged before and after insertion of our Trojan model. The table also contains the number of configurable RO paths used when designing the respective defense ring oscillator mechanism.

TABLE I. LIST OF DIFFERENT BENCHMARK CIRCUITS EMBEDDED WITH RO PATH

Benchmark	No. of inputs	No. of configurable RO paths	Counter value before Trojan insertion	Counter value after Trojan insertion	% change
c880	60	24	4346	4234	1.12
c2670	233	43	29e3	29b1	3.23
c3540	50	8	3581	3796	2.15
c5315	178	58	546d	542b	4.21
c6288	32	12	8042	8039	0.09
c7550	207	37	4a26	4a53	5.63

The output of each benchmark circuit is connected to an OR gate which is triggered by the counter as explained in the earlier section. We used a simple circuit to demonstrate our Trojan model insertion to make the circuit malfunction at a particular instance.

In our proposed Trojan model, the OR gate is the payload and is required to set the output to a value for a specific input condition for ADDER 4 circuit. This event is triggered by the counter as shown in Fig. 4. The counter is designed and simulated using Xilinx ISE. The simulation results are shown in Fig. 5. As seen in the figure, for a specific input condition '0001', the output of ADDER 4 is '1111' all the time.

We then performed the simulation by embedding a ring oscillator throughout the ADDER 4 design. The results show that our Trojan model was undetected in the presence of the ring oscillator circuitry, thus being successful in breaching this defense mechanism. The simulation results under Trojan-free and Trojan-inserted conditions prove that the output malfunction in the presence of ring oscillator monitoring scheme, thus making the defense method vulnerable to our Trojan model. Even in the scope of ASIC, dynamically configured ROs cannot guarantee the security of the design, since a clever attacker can always try to bypass them. Even if all paths (and all gates) of the circuit are covered by some ROs, the Trojan can still be inserted without significantly affecting the delay.

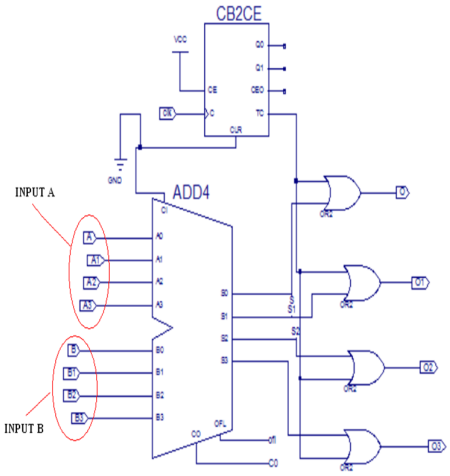

Figure 4. Schematic representation of Trojan circuit including a counter and OR gate

The trigger condition of the Trojan only adds load capacitance to some internal nodes of the circuit, which can be chosen from different ROs.

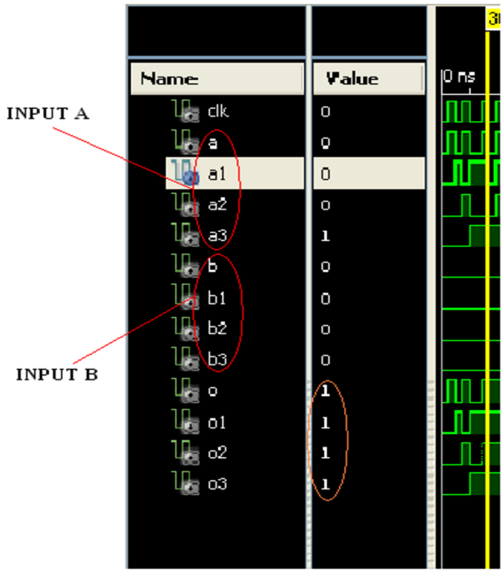

Figure 5. Behavioral Simulation results showing the output of ADD 4 malfunctioned as 1111. The adder 4 normally results into 0001 for given inputs of A (0001) and B (0000).

VI. Proposed Trojan Detection

Since the Trojan insertion was made at the RTL level, we can use Xilinx tools for analysis of the Circuit under Test. We term the entire design including the Trojan model as the Beta circuit. We performed a comprehensive analysis of this Beta circuit using Xilinx PlanAhead software (specifically Floorplanner utility). Such a beta design can easily evade a number of ring oscillator loops embedded into any IC design [11] where the output frequency for each ring oscillator of beta design will remain the same since the added components are located outside the ring.

The changes made in the design resulted in modified Placement and Routing (P&R) of the design. The P&R of Xilinx tool has been completely automated in order to provide designer with high design productivity and thus, give the designer more control. We could easily view the addition of Trojan components (indicated as white square dotted portion in Fig. 6), which is connected to the output ports of the Design under Test circuit as indicated. This led to successful detection of the proposed Trojan model.

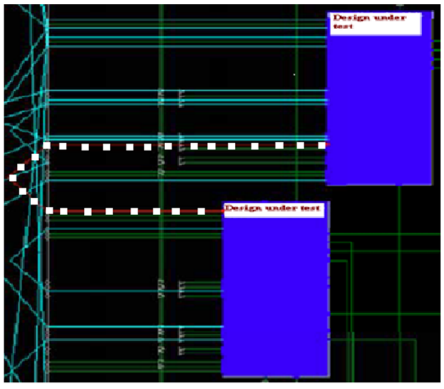

Figure 6. Place and route output of Beta design. The dotted portion illustrates Trojan components placement and routing.

VII. Conclusion And Future Scope

In this paper, we have demonstrated that a Trojan model that is placed specifically just before the output of any circuit was successful in evading the ring oscillator defense mechanism. The proposed model can be

redesigned stealthily in an appropriate manner in modern VLSI circuits to make its presence almost negligible. We also demonstrated by using Xilinx Floorplanner tool to analyze the P&R property of the circuit design and results show that our Trojan model can be successfully detected. Further research can be conducted by placing and routing the Trojan components in a manner that it can evade multiple defense schemes. Other work also includes low power design of such Trojan model that can be make detection more difficult when multiple defense methods are activated during testing phase. Consequently, further research should be focused on devising new methods that can help prevent Trojans that can cause considerable damage to critical systems and its applications.

REFERENCES

[1] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection, IEEE Design and Test of Computers, pp.10-25, 2010.

[2] D. Agrawal, S. Baktir, D. Karakoyunlu, P.Rohatgi, and B. Sunar, "Trojan Detection using IC Fingerprinting," in IEEE Symposium on Security and Privacy (SP), pp. 296–310, 2007.

[3] J. Li and J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection," IEEE Int. Hardware-Oriented Security and Trust, 2008

[4] J. Rajendran, V. Jyothi, O. Sinanoglu, and R. Karri, "Design and analysis of ring oscillator based Design-for-Trust technique". IEEE VLSI Test Symposium (VTS), PP. 105-110, 2011.

[5] Suh, G.E., Devadas, S.: Physical Unclonable functions for device authentication and secret key generation. In: Design Automation Conference, pp. 9{14. ACM Press, New York, NY, USA (2007)

[6] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware Trojan Detection and Isolation using Current Integration and Localized Current Analysis," in IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems (DFTVS08), pp. 87-95, 2008.

[7] Y. Jin and Y. Makris, "Hardware Trojan Detection using Path Delay Fingerprint," IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 51-57, 2008.

[8] J. Li and J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection," IEEE Int. Hardware-Oriented Security and Trust, 2008

[9] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy Hardware: Identifying and Classifying Hardware Trojans," IEEE Computer Magazine, pp. 39-46, Oct. 2010.

[10] Iscas-85 high level models web site. http://www.eecs.umich.edu/jhayes/iscas.restore/benchmark.html

[11] Andrew Ferraiuolo, Xuehui Zhang, and Mohammad Tehranipoor, "Experimental Analysis of a Ring Oscillator Network for Hardware Trojan Detection in a 90nm ASIC", International Conference on Computer-Aided Design, 2012

[12] X. Zhang and M. Tehranipoor, "RON: An on- chip ring oscillator network for hardware Trojan detection", DATE, 2011.

[13] J. Rajendran, V. Jyothi, O. Sinanoglu., and R. Karri, "Design and analysis of ring oscillator based Design-for-Trust technique", VTS, 2011

[14] A.Maiti, J. Casarona, L. Mchale, and P. Schaumont, "A large scale characterization of RO-PUF," IEEE International Symposium on Hardware Oriented Security and Trust, pp. 94-99, Jun. 2010.